IN THE CLAIMS

Please withdraw Claims 23-42 without prejudice, amend Claims 1 and 17, and add new Claims 43-53 as follows:

1. (Currently amended)   A method of formatting a plurality of input strings of data derived from respective ones of a plurality of data processors, comprising:

initializing the data in each of said strings;

finding the differences between said strings; and

providing said differences in a display to a user;

wherein said act of finding the differences comprises:

identifying groups of said data within said strings that are identical across said plurality of input strings; and

identifying groups of said data within said strings that appear in the same order within all of said strings.

2. (Previously presented)   The method of Claim 1, wherein the act of initializing comprises:

creating a symbol table having a plurality of symbol numbers associated therewith;

creating said symbol array having at least one element for each of said strings;

for each of said input strings, determining whether said each string is present in said symbol table; and

if said each string is present in said symbol table, obtaining at least one symbol number for said string from said symbol table.

3. (Previously presented)   A method of analyzing a plurality of strings of data derived from at least one data processing device, comprising:

initializing said data, said act of initializing including creating a symbol array;

analyzing said strings of data based at least in part on said symbol array; and

identifying at least one relationship between one or more of said data within one or more of said strings;

wherein said act of identifying comprises:

identifying groups of said data within said strings that are identical across said plurality of input strings; and

identifying groups of said data within said strings that appear in the same order within all of said strings.

4. (Previously presented)   The method of Claim 3, wherein the act of initializing comprises:

creating a symbol table having a plurality of symbol numbers associated therewith;

creating said symbol array having at least one element for each of said strings;

for each of said input strings, determining whether said each string is present in said symbol table; and

if said each string is present in said symbol table, obtaining at least one symbol number for said string from said symbol table.

5.-10. (Cancelled)

11. (Previously presented)   A storage device, comprising:

a computer readable medium;

a computer program stored on said computer readable medium, said program being adapted for analyzing a plurality of strings of data derived from at least one data processing device according to the method comprising:

initializing said data, said act of initializing including creating a symbol array;

analyzing said strings of data based at least in part on said symbol array; and

identifying at least one relationship between one or more of said data within one or more of said strings;

wherein said act of identifying comprises:

identifying groups of said data within said strings that are identical across said plurality of input strings; and

identifying groups of said data within said strings that appear in the same order within all of said strings.

12. (Previously presented)   A data processing device, comprising:

a processor adapted to process digital data and execute a computer program;

a storage device in data communication with said processor, said storage device comprising:

       a computer readable medium; and

       a computer program stored on said computer readable medium, said program being adapted for analyzing a plurality of strings of data derived from at least one data processing device according to the method comprising:

       initializing said data, said act of initializing including creating a symbol array;

       analyzing said strings of data based at least in part on said symbol array; and

       identifying at least one relationship between one or more of said data within one or more of said strings;

       wherein said act of identifying comprises:

           identifying groups of said data within said strings that are identical across said plurality of input strings; and

           identifying groups of said data within said strings that appear in the same order within all of said strings.

13. (Previously presented)   The device of Claim 12, further comprising a plurality of data interfaces adapted to receive said data strings from respective ones of a plurality of software processes running on respective ones of a plurality of data processors.

14. (Previously presented)   The device of Claim 13, further comprising said plurality of data processors.

15.-16. (Cancelled)

17. (Currently amended)   A method of evaluating the operation of a plurality of software processes running on respective ones of a plurality of digital processors, comprising:

       generating a first data string using a first of said plurality of software processes;

       generating a second data string using a second of said plurality of software processes;

       generating a third data string using a third of said plurality of software processes;

       inputting said first and second and third data strings into a debug software process;

analyzing, in a substantially contemporaneous process, said first and second and third data strings using said debug process; and

evaluating the operation of said processes based at least in part on said act of analyzing.

18. (Previously presented)   The method of Claim 17, wherein the act of analyzing comprises:

(i)   identifying common patterns within at least one first portion of said first and second and third data strings; and

(ii)   identifying differences within at least one second portion of said first and second and third data strings.

19. (Previously presented)   A multi-processor integrated circuit device, comprising:

a first processor core adapted to run a first software process, said first software process adapted to generate a first string of data;

a second processor core adapted to run a second software process said second software process adapted to generate a second string of data;

at least one data interface, wherein said first and second processors respectively transfer data comprising said first and second strings to an external debug process adapted to identify similarities and differences between the operation of said first and second software processes on said first and second processors, via said at least one interface;

wherein said debug process identifies the similarities and differences by:

identifying groups of said data within said first and second strings that are identical across at least both of said strings; and

identifying groups of said data within said first and second strings that appear in the same order within at least both of said strings.

20. (Previously presented)   The method of Claim 1, wherein said strings each comprise a plurality of lines, and said method further comprises:

forming a plurality of groups of lines, wherein a group comprises a sequence of lines that are the same in all of said strings; and

recursively analyzing, in order:

a first region of all the strings that appears before the first of said plurality groups;

each of a plurality of second regions occurring between two of said plurality of groups; and

a third region following the last of said plurality of groups.

21. (Previously presented)   The storage device of Claim 11, wherein said strings each comprise a plurality of lines, and said method further comprises:

forming a plurality of groups of lines, wherein a group comprises a sequence of lines that are the same in all of said strings; and

recursively analyzing, in order: a first region of all the strings that appears before the first of said plurality groups;

each of a plurality of second regions occurring between two of said plurality of groups; and

a third region following the last of said plurality of groups.

22. (Previously presented)   The integrated circuit device of Claim 19, wherein said strings each comprise a plurality of lines, and said debug process is further adapted to:

form a plurality of groups of lines, wherein a group comprises a sequence of lines that are the same in all of said strings; and

recursively evaluate, in order:

a first region of all the strings that appears before the first of said plurality groups;

each of a plurality of second regions occurring between two of said plurality of groups; and

a third region following the last of said plurality of groups.

23. (Withdrawn)   Computer storage apparatus having a media adapted to store a plurality of data, said data comprising a computer program, said computer program being adapted to process a plurality of input streams to identify groups of elements within said streams that are:

(i)   identical across said plurality of input streams; and

(ii)   which appear in the same order within all of said streams.

24. (Withdrawn)   The apparatus of Claim 23, wherein said input streams comprise a plurality of lines, and said program is further adapted such that if no groups of elements are identified, said program:

(i)   compares all of said streams and lines in order of occurrence; and

(ii)   provides a representation of each unique line prefixed by those streams that

contain said unique line.

25. (Withdrawn)   The apparatus of Claim 23, wherein said input streams comprise a

plurality of lines, and said program is further adapted such that if one or more groups are

identified, said program recursively analyzes a first region in all the streams before a first of said

one or more groups.

26. (Withdrawn)   The apparatus of Claim 25, wherein when said program identifies

two or more groups, said program recursively analyzes the region between each of individual

sequential ones of said groups in order, and the region following a last of said two or more

groups.

27. (Withdrawn)   A method of processing a plurality of input streams of data

elements, said method comprising:

identifying groups of said elements within said streams that are identical across said

plurality of input streams; and

identifying groups of said elements within said streams that appear in the same order

within all of said streams.

28. (Withdrawn)   The method of Claim 27, wherein said elements of said input

streams comprise a plurality of lines, and said method further comprises, where no groups of

elements are identified:

comparing all of said streams and lines in order of occurrence; and

providing a representation of each unique line, said representation being prefixed by those

streams that contain said unique line.

29. (Withdrawn)   The method of Claim 27, wherein said input streams comprise a

plurality of lines, and said method comprises recursively analyzes a first region in all the streams

before a first of said one or more groups.

30. (Withdrawn)   The method of Claim 29, wherein when two or more groups are identified, recursively analyzing the region between each of individual ones of said groups in order, and also the region following a last of said two or more groups.

31. (Withdrawn)   The method of Claim 27, wherein said processing of a plurality of input streams comprises processing two or more streams derived from respective ones of digital processors.

32. (Withdrawn)   The method of Claim 30, wherein said processing of a plurality of input streams comprises processing two or more streams derived from respective ones of digital processors.

33. (Withdrawn)   A computerized method for analyzing a plurality of input streams having a plurality of elements arranged in lines, comprising:

   identifying any of said streams that have no lines in common with any other stream; and

   removing such streams from further analysis.

34. (Withdrawn)   The method of Claim 33, further comprising:

   forming one or more groups of lines, where a group comprises a sequence of lines that are the same in all of said input streams.

35. (Withdrawn)   The method of Claim 34, wherein said act of forming comprises at least discovering unique lines that occur exactly once and in every one of said input streams.

36. (Withdrawn)   The method of Claim 35, wherein said act of forming further comprises expanding at least one line surrounding each of said unique lines as long as said at least one line contains elements that match in all of said input streams.

37. (Withdrawn)   The method of Claim 34, further comprising verifying that all of said groups occur in the same order in all of said streams.

38. (Withdrawn)   The method of Claim 33, further comprising evaluating all of said lines in all of said streams to determine the number of times each element occurs in each stream.

39. (Withdrawn)   The method of Claim 38, further comprising:

   determining which of said plurality of streams have less than a predetermined portion of its elements appear in no other stream; and

discarding all streams other than those identified during said act of determining.

40. (Withdrawn)   The method of Claim 39, further comprising sequentially incrementing through each of the lines of all non-discarded streams, said act of incrementing further comprising:

identifying each distinct element occurring at the selected line number; and

identifying those streams containing that element.

41. (Withdrawn)   A computerized method for analyzing a plurality of input streams having a plurality of elements arranged in lines, said streams being derived from respective ones of digital processors, the method comprising:

identifying any of said streams that have no lines in common with any other stream;

removing such identified streams from further analysis;

using the remaining streams, forming a plurality of groups of lines, wherein a group comprises a sequence of lines that are the same in all of said remaining input streams; and

recursively analyzing, in order:

a first region of all the streams that appears before the first of said plurality groups;

each of a plurality of second regions occurring between two of said plurality of groups; and

a third region following the last of said plurality of groups.

42. (Withdrawn)   The method of Claim 33, wherein said act of recursively analyzing comprises producing additional ones of said groups in said first, second or third regions.

43. (New)   A method of formatting a plurality of inputs of data derived from respective ones of a plurality of data processors, comprising:

initializing the data in each of said inputs;

finding the differences between said inputs; and

providing said differences in a display to a user;

wherein said act of finding the differences comprises:

identifying at least one unique string within at least one of said inputs; and

splitting the inputs into a series of matching chunks and non-matching chunks, wherein the matching chunks contain the at least one unique string, and the non-matching chunks contain that portion of said inputs not part of said matching chunks.

44. (New)   The method of Claim 43, wherein said plurality of inputs comprises at least three inputs of data.

45. (New)   The method of Claim 44, further comprising examining said series of matching chunks to ensure that they occur in the same order for every input; and

performing a string-by-string comparison on those matching chunks that do not occur in the same order for every input.

46. (New)   The method of Claims 45, wherein said string-by-string comparison method comprises:

(i)   determining a percentage of strings of said matching chunks that occur in at least one other of said inputs;

(ii)   identifying those of said inputs which contain strings that exceed a pre-set threshold;

(iii)   identifying a set of said inputs which do not exceed said threshold of step (ii);

(iv) comparing a first string of a first input of said set with a first string of each of the other inputs of said set; and

(v) separating into groups those inputs of said set which match the first string and those which do not match.

47. (New)   The method of Claim 45, further comprising growing said series of matching chunks by at least absorbing surrounding strings which are identical in all inputs.

48. (New)   A storage device, comprising:

a computer readable medium;

a computer program stored on said computer readable medium, said program being adapted for analyzing a plurality of data inputs derived from at least one data processing device according to the method comprising:

initializing said data, said act of initializing including creating a symbol array;

analyzing said inputs based at least in part on said symbol array; and

identifying at least one relationship between one or more of said data within one or more of said inputs;

wherein said act of identifying comprises:

identifying at least one unique string; and

splitting the inputs into matching chunks and non-matching chunks, wherein the matching chunks contain the at least one unique string, and the non-matching chunks contain that portion of said inputs not part of said matching chunks.

49. (New)   A multi-processor integrated circuit device, comprising:

a first processor core adapted to run a first software process, said first software process adapted to generate a first series of data;

a second processor core adapted to run a second software process said second software process adapted to generate a second series of data; and

at least one data interface, wherein said first and second processors respectively transfer data comprising said first and second series to an external debug process adapted to identify similarities and differences between the operation of said first and second software processes on said first and second processors, via said at least one interface;

wherein said debug process identifies the similarities and differences present in said data series by:

identifying at least one unique string from said first and second series; and

splitting the first and second series into matching chunks and non-matching chunks, wherein the matching chunks contain the at least one unique string, and the non-matching chunks contain that portion of said series not part of said matching chunks.

50. (New)   A method of formatting a plurality of inputs of data derived from respective ones of a plurality of data processors, comprising:

initializing the data in each of said inputs; and

identifying at least one relationship between one or more of said data within one or more of said inputs, said act of identifying comprising:

creating a list of anchors containing strings that occur exactly once in every one of said inputs;

marking at least one of said strings as belonging to a group; and

determining whether a string occurring before said group is identical in each of said inputs.

51. (New)   The method of Claim 50, further comprising examining a string immediately after said group to determine if it is identical in each of said inputs.

52. (New)   A method evaluating a plurality of inputs of data derived from respective ones of a plurality of data sources, comprising:

examining each of said inputs to identify any data strings which occur only once in that input, said any data strings comprising unique strings;

comparing unique strings in each of said inputs to identify any of said unique strings that also comprise unique strings in all other inputs, such strings comprising shared unique strings;

splitting each of said inputs into a series of matching and non-matching chunks, said matching chunks comprising said shared unique strings, and said non-matching chunks comprising that portion of said inputs not comprising said shared unique strings;

examining said inputs to verify that said matching chunks occur in the same order in every one of said inputs; and

growing said matching chunks by at least absorbing surrounding strings which are identical in all of said inputs.

53. (New)   The method of Claim 52, wherein if said act of examining indicates that said matching chunks do not occur in the same order in every one of said inputs, the method further comprises:

comparing a first string of a first input with a first string of each of the other of said

plurality of inputs; and

separating said inputs into at least a first category comprising those inputs for which said first strings match, and a second category comprising those inputs for which said first strings do not match.